Advanced Networking: how to nat AWS traffic with ANY private IP

29 April 2022 - 6 min. read

Advanced Networking | AWS Transit Gateway | Hybrid Cloud | Landing Zone | NAT Gateway

Introduction

In the Hybrid Cloud model, there is a lot of complexity related to the network when we want to create a perfect ecosystem between the Cloud and the on-premises environment.

In particular, the network connection through a VPN site-to-site is very delicate since we can have different complications, for example, the overlap between the CIDRs of the environments.

Another problem could be that you need to establish a VPN connection and on the other side there are constraints about the VPC CIDR dimension, for example, they cannot reserve a /16 for a single connection but a smaller network like a /27. Or again, in a worst-case scenario, we can establish a connection with a /27 that doesn't belong to the same CIDR class of the VPC.

This last case is a real situation we had to handle during a VPN configuration between our AWS architecture and an on-premises environment.

The third part could not establish a connection with the address space of our VPC since it is too large (/16) and overlaps with another environment.

If we were in an on-premises environment, there would be no problems since it is possible to create a virtual network to nat the traffic directed to another environment behind another address space. But in the cloud, this is not possible by design.

So we had to choose a smaller network space that was suitable for both of us to use to nat our traffic to them and, at the same time, allow them to reach our applications located in different network spaces.

We don't spend too much time explaining the characteristics of the networking components. Here you can find how to configure your accounts with a transit gateway and the nat gateway centralized with some theoretical concepts. We consider this article as an extension of the above-mentioned blog post, so we recommend giving it a look for a deeper understanding of it.

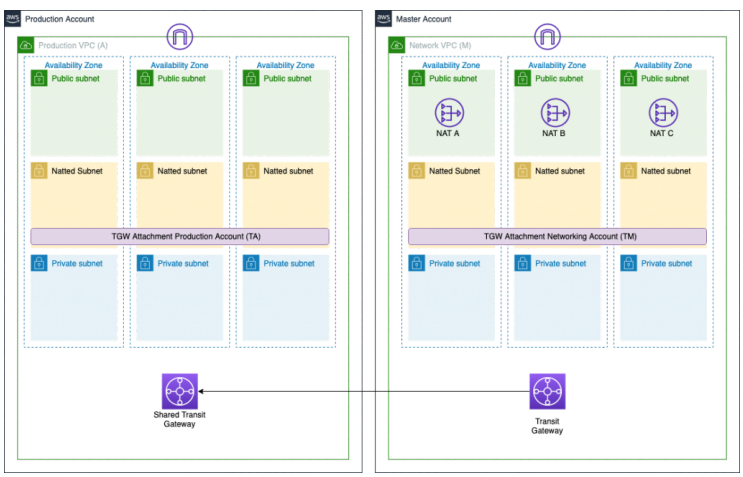
Architecture Design

For our use case, we are supposed to already have a small **landing zone** composed of two accounts, master and production, each one with its VPC linked through a transit gateway (the same architecture presented in the articles of the link below but for simplicity just with master account and production account).

The master account is responsible for managing the network configuration while in the production account we have our application environment. We want to establish a VPN connection with an on-premises environment that should accept and send traffic only to a /27 CIDR different from our VPC CIDR of master and production accounts.

What can we do? Let's begin with the fun part and design a solution for this use case!

We start from the network configuration illustrated below:



Here we have the two accounts presented before, connected with a Transit Gateway. We define two CIDR for our VPC, 10.100.0.0/16 for master account (VPC M) and 10.150.0.0/16 for production account (VPC A). The on-premises environment asks to establish a VPN connection with our accounts so we should establish the connection just with the master account and then use the transitivity characteristic of the transit gateway to connect also with the production account. The problem is that the on-premises environment already has a lot of VPN connections and the CIDR 10.100.0.0/16 is already in use and it is too large to establish a single VPN connection so we decided to use the CIDR 10.179.0.0/27 for it (it is important to choose the right size of the CIDR in order to avoid the risk of insufficient space of IPs)

Basically, we need to NAT behind this CIDR all the traffic coming from production and master accounts and directed to the VPN.

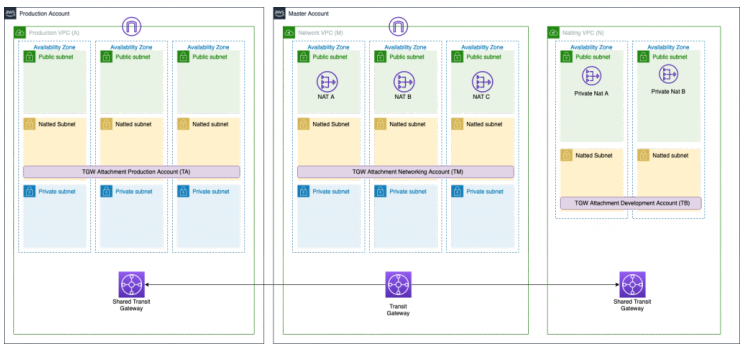
VPC Set up

We create a new VPC in the master account to use as a virtual network in order to nat the traffic directed to the VPN.

This VPC is an additional networking layer in front of the VPC M with the CIDR 10.179.0.0/16 (for simplicity we call it VPC N).

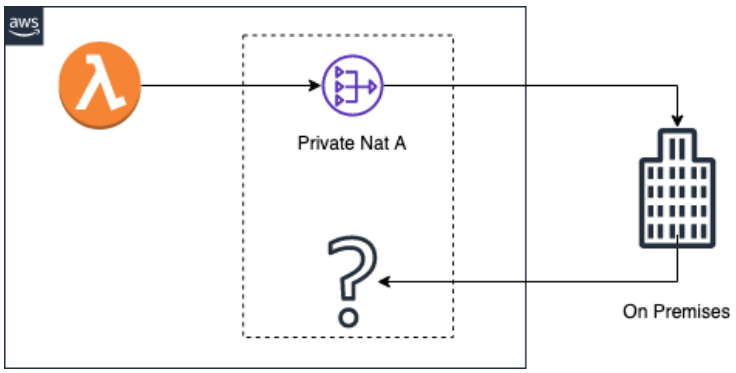
Note: you may be wondering why we haven't extended the VPC CIDR. Theoretically, we could have added an additional CIDR to the VPC and achieve the same result, **but** the difference from creating a new VPC is that the CIDR must belong to the same class as the primary CIDR: for example, if the CIDR is within the address space 10.x.x.x/8, you cannot use a CIDR within the other private address spaces defined by RFC1918 (192.168.x.x/16 and 172.16.x.x/12).

The architecture schema we realized is the following:



With the /27 we can create just two public subnets with the CIDR 10.179.0.0/28 and 10.179.0.16/28 and two natted subnets with the CIDR 10.179.0.32/28 and 10.179.0.48/28. The public subnets are those that can communicate with the on-premises environment while the natted subnets are necessary to redirect the traffic directed to the on-premises through the NAT. In the public subnets, we create the two private NAT gateway that will nat all the traffic coming from VPC M and VPC A.

Note that this is a source nat i.e. it is used only from the traffic coming from AWS environments, so the traffic coming from on-premises environment needs another way to communicate with our VPC A and M.



This image is a simple representation of the traffic path between the environments but at the moment only the application (lambda for example) can reach the on-premises and not vice versa.

Transit Routing

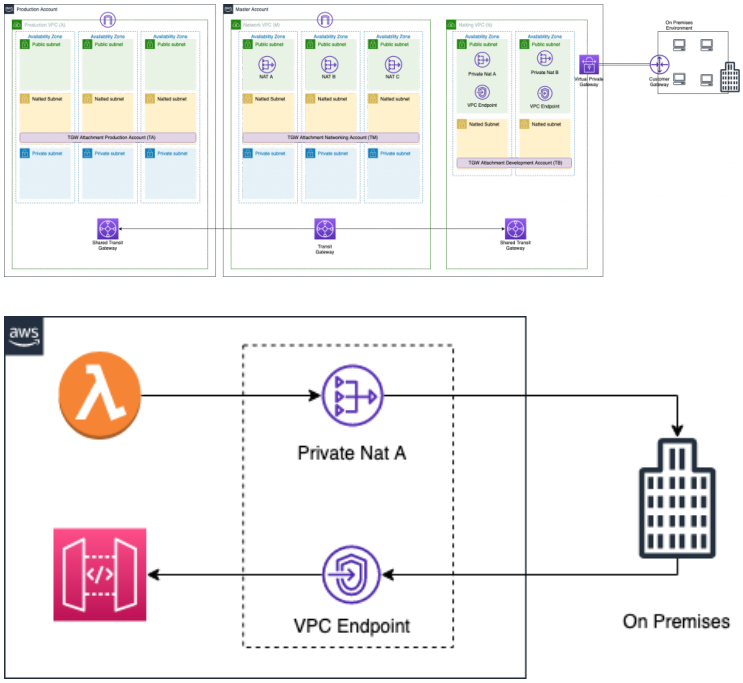
After creating the VPC, we need to configure the transit gateway in order to establish the connection between the existing VPCs and this new one. We create the transit gateway attachment in the natted subnet of the VPC N and we populate the transit gateway route table by adding the routes direct to the on-premises IPs through the attachment of VPC N.

VPN Set up

Now the last thing to do is to set up the VPN connection in the VPN N. We create a VPN site-to-site directly on the VPC N, so without attaching it to the transit gateway but to a virtual private gateway.

All is done! We can try to ping an on-premises machine from an ec2 inside VPC M or A and if all is ok we should see the ping from the IP of one of the two private nat gateway inside VPC N.

The final architecture should be this one.



As shown in the figure, based on the type of the application, the only thing that we can do is to create some VPC endpoints, for example for the API gateway. These will assume an IP inside 10.179.0.0/27 but the only protocol that we can use is HTTP/S.

Furthermore, we have a limited number of IPs so we have to keep in mind that we can create only a limited number of VPC endpoints.

To conclude

Even if coming up with this solution was not exactly a piece of cake - also due to fact that we didn't find any similar case on the web - we're pretty happy with it because it perfectly solved our this particular networking need.

The nice thing is the fact that for each VPN connection we need to establish we can build a specific VPC just to set up the connection even in case of particular constraints like in this case.

The other side of the coin is that this solution involves additional costs in the networking since we need to create an additional transit gateway attachment and 2 NAT gateways.

Furthermore, we need to create a VPC endpoint for each service that we need to contact from on-prem. So take in mind all these aspects before implementing this solution.

Do you know another way to solve this problem? Let us know in the comment!

Nicholas Farina

DevOps Engineer @ beSharp. I deal with the implementation and management of Cloud infrastructure on AWS.I am a CrossFit lover and in my free time, I go fishing mainly in salt water.

Simone Merlini

CEO and co-founder of beSharp, Cloud Ninja and early adopter of any type of * aaS solution. I divide myself between the PC keyboard and the one with black and white keys; I specialize in deploying gargantuan dinners and testing vintage bottles.
